Introduction to CTmeta: functions for lagged effects model parameters

R. M. Kuiper

```
library(CTmeta)
```

Introduction

This package started out as a package to perform meta-analysis on time-interval dependent discrete-time lagged effects parameter estimates (e.g., CLPM or VAR(1) estimates). This method makes use of the underlying continuous-time (CT) model and is therefore referred to as CTmeta. In the meantime, I included many more functionalities, namely more functions that can, for instance, transform, standardize, and plot the time-interval dependent discrete-time lagged effects parameters.

To be more specific, some of the functionalities are:

- Rendering plots of how the VAR(1) model lagged parameters (Phi) and residual covariance matrix vary as a function of the time interval (DeltaT). See the functions ‘PhiPlot’ and ‘SigmaVARPlot’, respectively.
- Determining the time-interval DeltaT for which each element of Phi(DeltaT) reaches its minimum or maximum. See the function ‘MaxDeltaT’.
- Rendering the time interval for which the VAR(1) model residual covariance matrix is a diagonal matrix. See the function ‘DiagDeltaT’.
- Rendering the standardized lagged effects from the unstandardized ones; or from cross-correlations and their covariance matrix. See the functions ‘StandPhi’ and ‘TransPhi_Corr’, respectively. These

functions also render the univariate (i.e., simultaneous) and multivariate (i.e., elliptical) confidence intervals of the (un)standardized lagged effects.

- Transforming (un)standardized VAR(1) model lagged parameters and residual covariance matrix as if another time-interval is used. See the function ‘StandTransPhi’. This function also renders the univariate (i.e., simultaneous) and multivariate (i.e., elliptical) confidence intervals of the (un)standardized lagged effects.
- Transforming VAR(1) model lagged parameters and residual covariance matrix into the (un)standardized ones of the CT(1) model and vice versa. See the functions ‘CTMparam’ and ‘VARparam’, respectively.

Below, I will give examples to demonstrate the use of these functions.

More details

More details about the methods and derivations can be found in:

- Kuiper, R. M., and Hamaker, E.L. (unpublished). Correlated residuals: What they do (not) represent.
- Kuiper, R. M., & Ryan, O. (2020). Meta-analysis of Lagged Regression Models: A Continuous-time Approach. Structural Equation Modeling, 27(3). https://doi.org/10.1080/10705511.2019.1652613

Installation and descriptions

```
# Install R package
# Note: Make sure you have Rtools (and a version which is compatible with your R version).
library(devtools)
install_github("rebeccakuiper/CTmeta")

# Load package
library(CTmeta)

# In case you use functions from this CTmeta package, please cite it:
citation("CTmeta")

# To look at the description of a function including example code, use ?functionname:
?Area
?ChecksCTM
?CTmeta
?CTMparam
?DiagDeltaT
?Gamma.fromCTM
?Gamma.fromVAR
?ggPhiPlot
?MaxDeltaT
?PhiPlot
?SigmaVARPlot
?StandPhi
?StandTransPhi
?TransPhi_Corr
?VARparam

# To obtain an overview of all functions in the package and their arguments:
lsf.str("package:CTmeta")
```

Usage

Area(DeltaT = 1, Phi, …)

ChecksCTM(Drift, Sigma)

CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR, …)

CTMparam(DeltaT, Phi, SigmaVAR, …)

DiagDeltaT <- function(Phi, SigmaVAR, …)

Gamma.fromCTM(Drift, Sigma)

Gamma.fromVAR(Phi, SigmaVAR)

ggPhiPlot(DeltaT = 1, Phi, …)

MaxDeltaT(DeltaT = 1, Phi, …)

PhiPlot(DeltaT = 1, Phi, …)

SigmaVARPlot(DeltaT = 1, Phi, SigmaVAR, …)

StandPhi(N = NULL, Phi, SigmaVAR, …)

StandTransPhi(DeltaTStar, DeltaT = 1, N = NULL, Phi, SigmaVAR, …)

TransPhi_Corr(DeltaTStar, DeltaT = 1, N = NULL, corr_YXYX, …)

VARparam(DeltaT = 1, Drift, Sigma, …)

Examples

Example 1: CTmeta

In this example, I will demonstrate CTmeta by making use of a simple example with S=3 primary studies. These three studies investigate the (cross-)lagged relationship between Stress and Anxiety (i.e., between q=2 variables), each rendering a 2x2 lagged effects matrix (Phi). All studies used different samples (and different sample sizes, N), but also collected the data with different time-intervals (DeltaT).

```
N <- matrix(c(643, 651, 473))
DeltaT <- matrix(c(2, 3, 1))
```

Note that the lagged effects matrix Phi varies with the time-interval DeltaT. If you would measure Stress and Anxiety every hour, Phi reflects the lagged relationship between current Stress and Anxiety levels and the Stress and Anxiety levels over one hour. This relationship is stronger than (different from) when you would measure every day (24 hours).

To combine the results, one should use meta-analysis. Because the lagged relationships are time-interval dependent, one needs CTmeta to account/correct for the different time intervals. Let us take that the interval of interest (DeltaTStar) is 1. Notable, DeltaTStar should preferable in the range of inspected time-intervals.

```
DeltaTStar <- 1
```

CTmeta needs as input, from all S=3 studies, the lagged effects matrix (Phi) and the residual covariance matrix (SigmaVAR) or the stationary covariance matrix (Gamma). These input matrices should stacked matrices of size S*q times q. I will use the example matrices stored in the package:

```
Phi <- myPhi
SigmaVAR <- mySigmaVAR
#Gamma <- myGamma # Note: CTmeta does not need both SigmaVAR and Gamma

# To see how a stacked matrix of size 3*2 x 2 looks like:
Phi
>      [,1] [,2]
> [1,] 0.25 0.10
> [2,] 0.20 0.36
> [3,] 0.35 0.20
> [4,] 0.30 0.46
> [5,] 0.15 0.00
> [6,] 0.10 0.26
```

Note that the CTmeta function will standardize these matrices (to make comparison and weighted averages of effects meaningful).

As with regular meta-analysis, one can choose between a fixed-effects and random-effects model and one can include moderators as well. Notably, the default is a fixed-effects model, but if one wants to generalize the results beyond the included studies, then one should use a random-effects model (‘FEorRE = 2’). In case one expects that the lagged effects matrices stacked in Phi are different / incomparable due to some study characteristics, then one should include these in the model. If, for example, some studies investigated the Stress-Anxiety relationship among students, while other studies did in a business context, one can include a moderator by including a dummy variable for this. Note that when the effect for this moderator is significant, then there is indeed a significant difference in the Stress-Anxiety relationship for these two subgroups.

Next, some code for performing different types of CTmeta-analyses, where only the output for the last CTmeta-analysis (a random-effects model with moderators) is shown.

```
### Example without moderators ###

## Fixed effects model ##
CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR)

## Random effects model ##
CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR, FEorRE = 2)


### Example with moderators ###
Mod <- matrix(c(64,65,47)) # 1 moderator
#Mod <- matrix(cbind(c(64,65,47), c(78,89,34)), ncol = q); colnames(Mod) <- c("Mod1", "Mod2") # two
         moderators, in each column 1

## Fixed effects model ##
CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR, Moderators = 1, Mod = Mod)


## Random effects model ##
CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR, Moderators = 1, Mod = Mod, FEorRE = 2)
> $DeltaTStar
> [1] 1
>
> $Overall_standPhi_DeltaTStar
>            [,1]       [,2]
> [1,] -1.0188556 -0.3134650
> [2,] -0.1552339 -0.8459448
> 
> $Overall_vecStandPhi_DeltaTStar
> overallPhi11 overallPhi12 overallPhi21 overallPhi22
>   -1.0188556   -0.3134650   -0.1552339   -0.8459448
> 
> $elliptical_CI
>    overallPhi11 overallPhi12 overallPhi21 overallPhi22
> LB   -1.9256654   -0.7884523   -0.5344532   -1.5771558
> UB   -0.1120457    0.1615223    0.2239853   -0.1147338
> 
> $alpha_CI
> [1] 0.05
> 
> $CovMx_OverallPhi_DeltaTStar
>              overallPhi11 overallPhi12 overallPhi21 overallPhi22
> overallPhi11  0.098728526 0.0142564447 0.0014498026  0.055719642
> overallPhi12  0.014256445 0.0342799046 0.0007401086  0.009963018
> overallPhi21  0.001449803 0.0007401086 0.0290818461  0.002727673
> overallPhi22  0.055719642 0.0099630184 0.0027276733  0.074873120
> 
> $tau2
> [1] 3.927205e-03 1.731663e-04 8.747329e-06 2.671946e-03
> 
> $messageTrans
> [1] "All eigenvalues are positive and real; hence, the Phi's are transformed to Phi(DeltaT*) to
       account for the time-interval dependency and standardized (to make comparison of effects
       meaningful)."
> 
> $messageMultivar
> [1] "For each study, the covariance matrix is positive definite; hence, a multivariate approach is
       used."
> 
> $StudiesComplexEV
> NULL
> 
> $StudiesNegEV
> NULL
> 
> $StudiesCovMxNotPosDef
> NULL
> 
> $ratioDeltaT
>          Ratio DeltaT*/DeltaT
> Study  1            0.5000000
> Study  2            0.3333333
> Study  3            1.0000000
> 
> $summaryMetaAnalysis
> 
> Multivariate Meta-Analysis Model (k = 12; method: ML)
> 
>   logLik  Deviance       AIC       BIC      AICc
>  25.2897    9.0800  -14.5793   -5.8510  669.4207
> 
> Variance Components:
> 
> outer factor: RandomPart (nlvls = 3)
> inner factor: overallPhi (nlvls = 4)
> 
>              estim    sqrt  k.lvl  fixed  level
> tau^2.1    0.0039  0.0627      3     no     11
> tau^2.2    0.0002  0.0132      3     no     12
> tau^2.3    0.0000  0.0030      3     no     21
> tau^2.4    0.0027  0.0517      3     no     22
> 
>     rho.11  rho.12  rho.21  rho.22    11  12  21  22
> 11       1  1.0000  1.0000  1.0000     -  no  no  no
> 12  1.0000       1  1.0000  1.0000     3   -  no  no
> 21  1.0000  1.0000       1  1.0000     3   3   -  no
> 22  1.0000  1.0000  1.0000       1     3   3   3   -
> 
> Test for Residual Heterogeneity:
> QE(df = 4) = 28.0320, p-val < .0001
> 
> Test of Moderators (coefficients 1:8):
> QM(df = 8) = 348.7735, p-val < .0001
> 
> Model Results:
> 
>                    estimate      se     zval    pval    ci.lb    ci.ub
> overallPhi11        -1.0189  0.3142  -3.2426  0.0012  -1.6347  -0.4030   **
> overallPhi12        -0.3135  0.1851  -1.6930  0.0904  -0.6763   0.0494    .
> overallPhi21        -0.1552  0.1705  -0.9103  0.3627  -0.4895   0.1790
> overallPhi22        -0.8459  0.2736  -3.0916  0.0020  -1.3822  -0.3096   **
> overallPhi11:Mod.    0.0248  0.0052   4.7256  <.0001   0.0145   0.0350  ***
> overallPhi12:Mod.    0.0066  0.0030   2.2035  0.0276   0.0007   0.0125    *
> overallPhi21:Mod.    0.0054  0.0027   1.9726  0.0485   0.0000   0.0108    *
> overallPhi22:Mod.    0.0234  0.0045   5.1632  <.0001   0.0145   0.0323  ***
> 
> ---
> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This results, among other things, in an overall 2x2 Phi matrix (if q=2). Although the varying time-intervals are corrected for, this overall Phi matrix is also dependent in the chosen time-interval (DeltaTStar). To obtain insight into this, one can make a Phi-plot of the resulting overall Phi, as is demonstrated in Section 1.1.

One of the main interests in lagged relationships is which of the variables is ‘causal dominant’, that is, has the highest predictive power. In this example, is current Stress a stronger predictor for Anxiety on the next time-interval of is current Anxiety for Stress on the next time-interval. This is of course also the case for the overall Phi. To evaluate the dominance of (overall) lagged effects one can use an AIC-type criterion called the GORICA (Altinisik, Nederhof, Hoijtink, Oldehinkel, Kuiper, conditionally accepted). This method is included in the restriktor package and is demonstrated in Section 1.2.

In the example above, the lagged effects matrices (stacked in Phi) were obtained from each primary study. It also possible to some or all studies report a (lagged) correlation matrix. Section 1.3 demonstrates how (lagged) correlation matrices can be used in CTmeta.

1.1: Phi-plot of resulting overall Phi

```
## Make customized Phi-plot of resulting overall Phi ##

# Extract the q times q overall Phi matrix
out_CTmeta <- CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR)
q <- sqrt(length(out_CTmeta$Overall_standPhi_DeltaTStar))
# resulting overall Phi:
overallPhi <- matrix(out_CTmeta$Overall_standPhi_DeltaTStar, byrow = T, ncol = q)

# Make Phi-plot:
Title <- as.list(expression(paste(Phi(Delta[t]), " plot:"),
      "How do the overall lagged parameters vary", "as a function of the time-interval"))
#
PhiPlot(DeltaTStar, overallPhi, Min = 0, Max = 40, Step = 0.5, Title = Title)
```

1.2: Evaluate dominance of overall Phi using GORICA

```
## Evaluate dominance of overall lagged effects matrix overallPhi ##

# Extract the vectorized overall standardized Phi matrix and its covariance matrix
out_CTmeta <- CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR)
est <- out_CTmeta$Overall_vecStandPhi_DeltaTStar
VCOV <- out_CTmeta$CovMx_OverallPhi_DeltaTStar

# Specify hypothesis
H1 <- "overallPhi12 < overallPhi21"
#H2 <- "overallPhi12 > overallPhi21"

# Evaluate dominance of cross-lagged using the GORICA
if (!require("restriktor")) install.packages("restriktor")
> Loading required package: restriktor
> This is restriktor 0.2-800
> restriktor is BETA software! Please report any bugs.
# Use restriktor package for function goric().
# Authors of goric(): Vanbrabant and Kuiper.
library(restriktor)
#goric(est, VCOV = VCOV, H1, H2, type = "gorica", comparison = "none")
# or equivalently:
goric(est, VCOV = VCOV, H1, type = "gorica", comparison = "complement")
> restriktor (0.2-800): generalized order-restriced information criterion approximation:
>
> Results:
>         model  loglik  penalty   gorica  gorica.weights
> 1           H1  12.395    3.500  -17.791           0.994
> 2  complement   7.235    3.500   -7.470           0.006
> ---
> The order-restricted hypothesis 'H1' has 174.222 times more support than its complement.
```

1.3: A (lagged) correlation matrix as input

```
## What if primary studies report a (lagged) correlation matrix ##

# Supose all S=3 primary studies reported the following lagged correlation matrix:
q <- 2
corr_YXYX <- matrix(c(1.00, 0.40, 0.63, 0.34,
                      0.40, 1.00, 0.31, 0.63,
                      0.63, 0.31, 1.00, 0.41,
                      0.34, 0.63, 0.41, 1.00), byrow = T, ncol = 2*q)

# In the example below, the same N and DeltaT(Star) values are used:
N <- matrix(c(643, 651, 473))
DeltaT <- matrix(c(2, 3, 1))
DeltaTStar <- 1

# Use the function 'TransPhi_Corr' to calculate the corresponding standardized lagged effects matrix
         per primary study.
# Note that one can already make the time-intervals equal via the arguments DeltaTStar and DeltaT,
         but CTmeta can as well.
# In this example, I deliberately make the time-intervals unequal,
# - such that the example is in line with the input (i.e., DeltaT <- matrix(c(2, 3, 1)))
# - such the resulting overall Phi should equal the Phi that underlies this lagged correlation
         matrix.
out_1 <- TransPhi_Corr(DeltaTStar = DeltaT[1], DeltaT = 1, N = N[1], corr_YXYX)
Phi_1 <- out_1$standPhi_DeltaTStar
SigmaVAR_1 <- out_1$standSigmaVAR_DeltaTStar
out_2 <- TransPhi_Corr(DeltaTStar = DeltaT[2], DeltaT = 1, N = N[2], corr_YXYX)
Phi_2 <- out_2$standPhi_DeltaTStar
SigmaVAR_2 <- out_2$standSigmaVAR_DeltaTStar
out_3 <- TransPhi_Corr(DeltaTStar = DeltaT[3], DeltaT = 1, N = N[3], corr_YXYX)
Phi_3 <- out_3$standPhi_DeltaTStar
SigmaVAR_3 <- out_3$standSigmaVAR_DeltaTStar

# Make Phi
Phi <- rbind(Phi_1, Phi_2, Phi_3) # This, returns a stacked matrix of size S q times q.
SigmaVAR <- rbind(SigmaVAR_1, SigmaVAR_2, SigmaVAR_3)
# For more details, see ?TransPhi_Corr

# Run CTmeta:
CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR)

# The overall q-times-q (here, 2x2) lagged effects matrix Phi
out_CTmeta <- CTmeta(N, DeltaT, DeltaTStar, Phi, SigmaVAR)
out_CTmeta$Overall_standPhi
```

Example 2: Time-interval dependency

2.1 Plots

This example focuses on the time-interval dependency of (cross-)lagged relationships for one study – and can thus also be used for the overall (cross-)lagged relationships resulting from CTmeta. Let us assume we investigate the (cross-)lagged relationship between Stress and Anxiety (i.e., between q=2 variables), represented by a 2x2 discrete-time lagged effects matrix Phi and a 2x2 discrete-time residual covariance matrix SigmaVAR (called Psi in Kuiper and Hamaker).

Note that both the lagged effects matrix Phi and the residual covariance matrix SigmaVAR vary with the time-interval DeltaT. If you would measure Stress and Anxiety every hour, Phi reflects the lagged relationship between current Stress and Anxiety levels and the Stress and Anxiety levels over one hour. Likewise, the residual covariance matrix SigmaVAR is the one that belongs to the every-hour lagged relationship. The lagged relationship vary with the chosen time-interval and for a long enough time-interval the lagged relationship is damped out, that is, the cross-lagged effects are zero (and the autoregressive effects are one). Then, the residual covariance matrix equals the stationary covariance matrix (Gamma) which equals the (contemporaneous) covariance matrix of the contemporaneous variables. To visualize this,there are functions that plot the elements of these matrices for a range of time-intervals:

```r
### Make Phi-plot ###

## Example 1 ##

# Phi(DeltaT)
DeltaT <- 1
Phi <- myPhi[1:2,1:2]

# Example 1.1: unstandardized Phi #
PhiPlot(DeltaT, Phi)

# Example 1.2: standardized Phi #
SigmaVAR <- diag(2) # for ease
PhiPlot(DeltaT, Phi, Stand = 1, SigmaVAR = SigmaVAR)


## Example 2: input from fitted object of class "varest" ##

DeltaT <- 1
data <- myData
#if (!require("vars")) install.packages("vars")
library(vars)
out_VAR <- VAR(data, p = 1)

# Example 2.1: unstandardized Phi #
PhiPlot(DeltaT, out_VAR)

# Example 2.2: standardized Phi #
PhiPlot(DeltaT, out_VAR, Stand = 1)


## Example 3: Change plot options ##
# Note: use Phi or Drift from Example 1
q <- dim(Phi)[1]
WhichElements <- matrix(1, ncol = q, nrow = q) # Now, all elements are 1
diag(WhichElements) <- 0 # Now, the autoregressive parameters are excluded by setting the diagonals
         to 0.
Lab <- c("12", "21")
Labels <- NULL
```

```
for(i in 1:length(Lab)){
 e <- bquote(expression(Phi(Delta[t])[.(Lab[i])]))
 Labels <- c(Labels, eval(e))
}
Col <- c(1,2)
Lty <- c(1,2)
# Standardized Phi
PhiPlot(DeltaT = 1, Phi, Stand = 1, SigmaVAR = SigmaVAR, Min = 0, Max = 10, Step = 0.05,
         WhichElements = WhichElements, Labels = Labels, Col = Col, Lty = Lty)

# Note that you can also use 'ggPhiPlot'.
# Then, you can customize this plot like you would do with a regular ggplot.


### Make Psi-plot/SigmaVAR-plot ###

# Phi(DeltaT)
DeltaT <- 1
Phi <- myPhi[1:2,1:2]
SigmaVAR <- diag(2) # for ease

# Example 1.1: unstandardized Phi&SigmaVAR #
SigmaVARPlot(DeltaT, Phi, SigmaVAR)

# Example 1.2: standardized Phi&SigmaVAR #
SigmaVARPlot(DeltaT, Phi, SigmaVAR, Stand = 1)

# Notes:
# Like in the Phi-plot, a 'varest' object can be used.
# Like in the Phi-plot, the plot can be customized (but there is no ggplot variant)
```

One can also have the continuous-time matrices: the lagged effects matrix Drift and the residual covariance matrix Sigma (also calles diffusion matrix). Based on those matrices, one can also make a Phi- and SigmaVAR-plot:

```
# I now calculate the continuous-time equivalent of Phi, that is, the underlying drift matrix
if (!require("expm")) install.packages("expm") # Use expm package for function logm()
> Loading required package: expm
> Loading required package: Matrix
>
> Attaching package: 'expm'
> The following object is masked from 'package:Matrix':
>
>     expm
library(expm)
Drift <- logm(Phi)/DeltaT

# Phi-plot: unstandardized Drift/Phi #
PhiPlot(DeltaT, Drift = Drift, Min = 0, Max = 10, Step = 0.01)

# Phi-plot: standardized Drift/Phi #
SigmaVAR <- diag(2) # for ease
PhiPlot(DeltaT, Phi, Stand = 1, SigmaVAR = SigmaVAR)
```

```
# SigmaVAR-plot: unstandardized Drift&Sigma / Phi&SigmaVAR #
Sigma <- diag(2) # for ease. Note that this is not the CT-equivalent of SigmaVAR.
SigmaVARPlot(DeltaT, Drift = Drift, Sigma = Sigma, Min = 0, Max = 10, Step = 0.01)
```

2.1.1 Area under curves in Phi-plot

The area under a curve is the magnitude of the displacement, which is equal to the distance traveled (only for constant acceleration). As a comparison, when the plot would show the variation of a drug concentration in blood plasma as a function of time, the area under the curve (from zero to infinity) represents the total drug exposure across time. Such a measure might be interesting when comparing lagged effects matrices using different formulations (e.g., in the drugs example, capsule vs tablet of same dose), but may also reflect which variable has overall more predictive strength – further research is needed to obtain more insight in the relevance of this measure.

Code to calculate the area under the curve for each of the elements in Phi:

```
Area(DeltaT, Phi)
> $Area
>           [,1]     [,2]
> [1,] 0.7699825 0.244806
> [2,] 0.4896121 1.039269
> 
> $Area_range
>           [,1]     [,2]
> [1,] 0.7699825 0.244806
> [2,] 0.4896121 1.039269

# If, for instance, the time-interval range from 1 to 2 should be inspected (and not 0 to infinity),
         then use:
Area(DeltaT, Phi, t_min = 1, t_max = 2)
> $Area
>           [,1]     [,2]
> [1,] 0.7699825 0.244806
> [2,] 0.4896121 1.039269
> 
> $Area_range
>           [,1]       [,2]
> [1,] 0.1480669 0.08153651
> [2,] 0.1630730 0.23775709

# Notes:
# One can also use a fitted object of the classes "varest" and "ctsemFit".
# One can also use the Drift matrix.
```

2.1.2 Maximum or minimum of curves in Phi-plot

To calculate for each element in Phi what the optimum is, thus either its maximum or minumum, one can use the following function:

```
MaxDeltaT(DeltaT, Phi)
> $DeltaT_MinOrMaxPhi
>            [,1]       [,2]
> [1,] 22.2610551  0.7992313
> [2,]  0.7992313 22.9205743
> 
> $MinOrMaxPhi
>              [,1]         [,2]
> [1,] 8.459562e-09 1.025501e-01
> [2,] 2.051002e-01 1.078231e-08

# Notes:
# One can also use a fitted object of the classes "varest" and "ctsemFit".
# One can also use the Drift matrix.
```

2.2 Transformations

2.2.1 Discrete-time <-> continuous-time

There are two types of lagged effects models: the discrete-time (DT) and continuous-time (CT) models. Both are related (for more details see the references at the top). There are functions to transform the one to the other:

```
### From DT to CT ###

## Example 1 ##
Phi <- myPhi[1:2, 1:2]
SigmaVAR <- diag(2) # for ease
DeltaT <- 1
#
CTMparam(DeltaT, Phi, SigmaVAR)
> $eigenvalueDrift
>           [,1]       [,2]
> [1,] -1.875619  0.0000000
> [2,]  0.000000 -0.7836412
> 
> $eigenvaluePhi_DeltaT
> [1] 0.4567399 0.1532601
> 
> $StableProcess
> [1] FALSE
> 
> $Drift
>            [,1]       [,2]
> [1,] -1.5275304  0.3598189
> [2,]  0.7196378 -1.1317296
> 
> $Sigma
>            [,1]       [,2]
> [1,]  3.2370355 -0.9260082
> [2,] -0.9260082  2.5960426
> 
> $Gamma
>           [,1]      [,2]
> [1,] 1.0855262 0.1102123
> [2,] 0.1102123 1.2170170
```

```
>
> $standDrift
>            [,1]       [,2]
> [1,] -1.5275304  0.3809887
> [2,]  0.6796507 -1.1317296
>
> $standSigma
>            [,1]       [,2]
> [1,]  2.9819968 -0.8056499
> [2,] -0.8056499  2.1331194
>
> $standGamma
>            [,1]       [,2]
> [1,] 1.00000000 0.09588739
> [2,] 0.09588739 1.00000000

# Notes:
# One can also use a fitted object of the class "varest".
# One can also use the Gamma matrix instead of SigmaVAR.


### From CT to DT ###

# Create matrices from DT process - normally, you would have these matrices
CTM <- CTMparam(DeltaT, Phi, SigmaVAR)
Drift <- CTM$Drift
Sigma <- CTM$Sigma

DeltaT <- 1
VARparam(DeltaT, Drift, Sigma)

# Notes:
# One can also use a fitted object of the class "ctsemFit".
# One can also use the Gamma matrix instead of Sigma.
```

2.2.2 Gamma

For both the DT and CT model, it holds that there are three types of matrices: 1) the lagged effects matrix (Phi or Drift), 2) the residuals covariance matrix (SigmaVAR or Sigma), and 3) the stationary covariance matrix (Gamma - this is the same for both models, which makes sense since it is the covariance matrix of the contemporaneous variables). These three types of matrices are related: when you know two of them, the other one can be calculated. Next, the code for two functions that can calculate Gamma from the DT and CT models, respectively:

```
# Using DT matrices
Gamma.fromVAR(Phi, SigmaVAR)
>           [,1]      [,2]
> [1,] 1.0855262 0.1102123
> [2,] 0.1102123 1.2170170

# Using CT matrices
Gamma.fromCTM(Drift, Sigma)
>           [,1]      [,2]
> [1,] 1.0855262 0.1102123
> [2,] 0.1102123 1.2170170
```

2.2.3 Checks

When you have the CT matrices, you can also do checks on these matrices. For example, the covariance matrices should be positive definite. These checks can be done via:

```
ChecksCTM(Drift, Sigma)
> $ChecksAreFine
> [1] TRUE
> 
> $Gamma
>           [,1]      [,2]
> [1,] 1.0855262 0.1102123
> [2,] 0.1102123 1.2170170
> 
> $error
> [1] "All matrices (Drift, Sigma, and Gamma) are fine."

# Note: One can also use a fitted object of class "ctsemFit"
```

2.2.4 Standardization

One can also obtain the standardized estimates from the unstandardized ones. It is also possible to obtain the multivariate (elliptical) confidence intervals (CIs) for the lagged effects estimates. Note that most software render the univariate ones, thus not taking into account the covariances between the estimates. The code to obtain this is given next:

```
## Obtain only standardized lagged effects ##
StandPhi(N = NULL, Phi, SigmaVAR)
> $StandPhi_DeltaT
>           [,1]      [,2]
> [1,] 0.2500000 0.1058835
> [2,] 0.1888869 0.3600000
> 
> $standSigmaVAR_DeltaT
>           [,1]      [,2]
> [1,] 0.9212122 0.0000000
> [2,] 0.0000000 0.8216812
> 
> $standGamma
>            [,1]       [,2]
> [1,] 1.00000000 0.09588739
> [2,] 0.09588739 1.00000000
# or
#StandPhi(Phi = Phi, SigmaVAR = SigmaVAR)

## Obtain standardized lagged effects and multivariate CIs ##
# In that case, input for the sample size N is needed as well.
N <- 643
StandPhi(N, Phi, SigmaVAR)
```

```
> $StandPhi_DeltaT
>           [,1]      [,2]
> [1,] 0.2500000 0.1058835
> [2,] 0.1888869 0.3600000
> 
> $vecStandPhi_DeltaT
> [1] 0.2500000 0.1058835 0.1888869 0.3600000
> 
> $CovMx_vecStandPhi_DeltaT
>               [,1]          [,2]          [,3]          [,4]
> [1,]  0.0014504849 -0.0001390832  0.0000000000  0.0000000000
> [2,] -0.0001390832  0.0014504849  0.0000000000  0.0000000000
> [3,]  0.0000000000  0.0000000000  0.0012937694 -0.0001240562
> [4,]  0.0000000000  0.0000000000 -0.0001240562  0.0012937694
> 
> $multiCI_vecStandPhi_DeltaT
>        Phi11      Phi12     Phi21    Phi22
> LB 0.1631628 0.01904629 0.1068749 0.277988
> UB 0.3368372 0.19272065 0.2708989 0.442012
> 
> $standSigmaVAR_DeltaT
>           [,1]      [,2]
> [1,] 0.9212122 0.0000000
> [2,] 0.0000000 0.8216812
> 
> $standGamma
>            [,1]       [,2]
> [1,] 1.00000000 0.09588739
> [2,] 0.09588739 1.00000000

# Notes:
# One can also use a fitted object of the classes "varest" and "ctsemFit".
# One can also use the Gamma matrix instead of Sigma.
```

2.2.5 Same time-interval

To make results from multiple studies comparable, it might be needed to transform the matrices (which are the matrices for the time-interval DeltaT) to the ones as if another time-interval (DeltaTStar) was used:

```
DeltaTStar <- 1
DeltaT <- 2

## Obtain only (unstandardized) transformed lagged effects ##
StandTransPhi(DeltaTStar, DeltaT, N = NULL, Phi)
# or
#StandPhi(Phi = Phi, SigmaVAR = SigmaVAR)

## obtain only (un)standardized transformed lagged effects ##
StandTransPhi(DeltaTStar, DeltaT, N = NULL, Phi, SigmaVAR)
# or
#StandTransPhi(DeltaTStar, DeltaT, Phi = Phi, SigmaVAR = SigmaVAR)

## Obtain (un)standardized transformed lagged effects and multivariate CIs ##
```

```
# In that case, input for the sample size N is needed as well.
N <- 643
StandTransPhi(DeltaTStar, DeltaT, N, Phi, SigmaVAR)

# Notes:
# One can also use a fitted object of the classes "varest" and "ctsemFit".
# One can also use the Gamma matrix instead of Sigma.
```

Example 3: Correlated residuals

As elaborated in Kuiper and Hamaker (unpublished), even though the focus in lagged effects models is the strength and sign of the lagged relationships, the residuals are of interest as well. As demonstrated above, via the SigmaVAR-plot (Psi-plot), the residual covariance matrix of the discrete-time model varies with the chosen time-interval, like the lagged relationships (in Phi) do. As stated above, the code to procude such a SigmaVAR-plot is given by:

```
### Make Psi-plot/SigmaVAR-plot ###

# Phi(DeltaT)
DeltaT <- 1
Phi <- myPhi[1:2,1:2]
SigmaVAR <- diag(2) # for ease

# Example 1.1: unstandardized Phi&SigmaVAR #
SigmaVARPlot(DeltaT, Phi, SigmaVAR)

# Example 1.2: standardized Phi&SigmaVAR #
SigmaVARPlot(DeltaT, Phi, SigmaVAR, Stand = 1)

# Notes:
# Like in the Phi-plot, a 'varest' object can be used.
# Like in the Phi-plot, the plot can be customized (but there is no ggplot variant)
```

Since SigmaVAR varies with the chosen time-interval DeltaT, there may exist a DeltaT for which SigmaVAR is diagonal. In that case, the discrete-time residuals are uncorrelated. Note that for DeltaT = 0, SigmaVAR is diagonal, but there may be more positive time-intervals for which SigmaVAR is diagonal. The code to calculate this DeltaT (if it exists) is:

```
# Calculate DeltaT for which SigmaVAR is diagonal
DiagDeltaT(Phi, SigmaVAR = SigmaVAR)
> $DeltaT_diag
> [1] 1
>
> $message
> [1] "No message / warning / error. Hence, there is positive DeltaT for which the
        diagonals/variances in SigmaVAR are positive (and off-diagonals 0)."
>
> $Phi_DeltaT_diag
>      [,1] [,2]
> [1,] 0.25 0.10
> [2,] 0.20 0.36
```

```
> 
> $SigmaVAR_DeltaT_diag
>      [,1] [,2]
> [1,]    1    0
> [2,]    0    1
> 
> $Gamma
>           [,1]      [,2]
> [1,] 1.0855262 0.1102123
> [2,] 0.1102123 1.2170170
> 
> $StandPhi_DeltaT_diag
>           [,1]      [,2]
> [1,] 0.2500000 0.1058835
> [2,] 0.1888869 0.3600000
> 
> $StandSigmaVAR_DeltaT_diag
>           [,1]      [,2]
> [1,] 0.9212122 0.0000000
> [2,] 0.0000000 0.8216812
> 
> $Gamma_s
>            [,1]       [,2]
> [1,] 1.00000000 0.09588739
> [2,] 0.09588739 1.00000000
> 
> $errorMatrices
> [1] "All matrices (Drift, Sigma, and Gamma) are fine."
> 
> $message_startvalues
> [1] "In case the Psi-plot/SigmaVAR-plot does show a solution (or another solution) for DeltaT such
        that Psi is diagonal (i.e., the covariances are 0), \n\n  alter the starting value for
        'DeltaT_diag'. Notably, by default, the value 1 is used."
```

```
# Notes:
# - The function 'SigmaVARPlot' can help to see whether there is a DeltaT for
# which SigmaVAr(DeltaT) is diagonal.
# One can then alter the starting value of DeltaT ('xstart_DeltaT') if needed.
# - A 'varest' object can be used as well.
```

Even though this function calculates the DeltaT for which SigmaVAR is diagonal, it may not be that usefull. Kuiper and Hamaker show that correlated discrete-time residuals are supposed to be an indication for omitted common causes (or an effect at a shorter time-interval), but it actually is not. It also does not signal that lagged effects relationships are distorted, since omitted unique causes will not effect the residual correlations. It may be better to inspect continuous-time residuals, by looking at Sigma. Correlated continuous-time residuals signal one or more omitted relevant variables (so, common or unique omitted causes). The correlated continuous-time residuals then warn us that the found lagged effects do not reflect the causal relationships, but the predicting relationships. Unfortunately, it is not a measure for the extent of the distortion. As stated above, code to obtain Sigma from the discrete-time parameter matrices is given by:

```
### From DT to CT ###
```

```
## Example 1 ##
Phi <- myPhi[1:2, 1:2]
SigmaVAR <- diag(2) # for ease
DeltaT <- 1
#
CTM <- CTMparam(DeltaT, Phi, SigmaVAR)
CTM$Sigma
>             [,1]       [,2]
> [1,]  3.2370355 -0.9260082
> [2,] -0.9260082  2.5960426

# Notes:
# One can also use a fitted object of the class "varest".
# One can also use the Gamma matrix instead of SigmaVAR.
```